Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of Claims:

1. (currently amended) A method of processing a query in a system in an object oriented programming environment, comprising:

~~encoding an access request statement from an application into a structured query language statement with a JDBC;~~

using a [[code]] base class to create a condition filter in [[the]] a standard query language statement, the condition filter defining properties to be satisfied by a result of [[the]] a query, ~~and the condition filter using an object to execute a precompiled query language statement~~, wherein a portion of data values in the condition filter are each replaced by a placeholder and a corresponding data value list for the placeholders is created using a tree data structure to store conditions;

precompiling the standard query language statement including the condition filter;

storing the precompiled standard query language statement;

receiving the query from an application including values for each of the data values represented by placeholders; and

using the precompiled standard query language statement including the received values from the application to request the result of the query from a database;

repeating the receiving and the using for another query, wherein the precompiled query language statement is executed multiple times without being recompiled. ~~; and~~]]

~~sending the standard query language statement to a database.~~

2. (currently amended) The method of processing a query according to claim 1, wherein ~~data values in the condition filter are replaced with~~ the placeholder is a question mark character ~~and a corresponding data value list is~~

~~created~~.

3. (canceled)

4. (currently amended) The method of processing a query according to claim 1, wherein the [[code]] condition filter includes LIKE, AND, and OR operators.

5. (currently amended) The method of processing a query according to claim 1, wherein the [[code]] condition filter includes one of IS NULL and IS NOT NULL functions.

6. (currently amended) The method of processing a query according to claim 1, wherein the [[code]] condition filter includes one of UPPER, LOWER, and INITCAP functions.

7. (currently amended) The method of processing a query according to claim 1, wherein the [[code]] condition filter comprises TO_DATE function.

8. (currently amended) The method of processing a query according to claim 1, wherein the [[code]] condition filter has parameters and none of the parameters is null.

9. (currently amended) A method of processing a query in a system in an object oriented programming environment, comprising:

~~encoding an access request statement from an application into a structured query language statement with a JDBC;~~

using an application programming interface (API) to create a standard query language (SQL) WHERE clause statement in the SQL statement and to pass the SQL WHERE clause statement to a persistent object framework (POF), wherein the API uses a tree data structure for storing conditions wherein data values of unknown parameters in the SQL WHERE clause statement are replaced with a question mark character and a corresponding data value list is created; [[and]]

receiving a query from an application;

sending the SQL WHERE clause statement to the database,

wherein the SQL WHERE clause statement includes a condition filter and uses a PreparedStatement object that uses the tree data structure to set values of the unknown parameters unknown, and wherein the SQL WHERE clause statement is executed multiple times without being recompiled.

10. (canceled)

11. (canceled)

12. (currently amended) An application programming interface (API) for a database query system in an object oriented programming environment, the application programming interface adapted to effect the steps comprising:

~~receiving a structured query language statement generated by a JDBC from an access request statement of an application;~~

creating a condition filter for a standard query language (SQL) WHERE clause statement; and

passing the condition filter to a persistent object framework,

wherein the SQL WHERE clause statement uses a PreparedStatement object to request a query, and wherein the query is executed multiple times without being recompiled.

13. (previously presented) The API according to claim 12, wherein data values in the condition filter for a SQL WHERE clause statement are replaced with a question mark character and a corresponding data value list is created.

14. (currently amended) A computer program product comprising a computer useable medium having computer readable code embodied therein for a database query, the computer program product adapted to effect the steps comprising:

~~encoding an access request statement from an application into a structured query language statement with a JDBC;~~

making a connection with a database;

~~using a code to create~~ creating a condition filter in a standard query language statement, the condition filter defining properties to be satisfied by a result of [[the]] a query, and the condition filter using an object to execute a precompiled query language statement, wherein the query language statement is executed multiple times without being recompiled; and

[[sending]] using the standard query language statement to request the result of the query from the database, wherein the using comprises responding to an access request from an application, the access request including values of parameters implemented in the standard query language statement by the object.

15. (currently amended) The computer program product according to claim 14, wherein data values in the condition filter are replaced with a question mark character and a corresponding data value list is created, data value list being used to implement the parameter values from the application.

16. (currently amended) The computer program product according to claim 14, wherein ~~the code includes~~ a tree data structure is used for storing conditions, the tree data structure being accessible by the executing object for the condition filter to set the parameters with the parameter values.

17. (currently amended) The computer program product according to claim 14, wherein the [[code]] condition filter includes LIKE, AND, and OR operators.

18. (currently amended) The computer program product according to claim 14, wherein the [[code]] condition filter includes one of IS NULL and IS NOT NULL functions.

19. (currently amended) The computer program product according to claim 14, wherein the [[code]] condition filter includes one of UPPER, LOWER, and INITCAP functions.

20. (currently amended) The computer program product according to claim 14, wherein the [[code]] condition filter includes TO_DATE function.

21. (currently amended) The computer program product according to claim 14, wherein the [[code]] condition filter has parameters and none of the parameters is null.

22. (currently amended) A computer program product comprising a computer useable medium having computer readable code embodied therein for a database query, the computer program product comprising:

~~means for encoding an access request statement from an application into a structured query language statement with a JDBC;~~

means for ~~using a code to create~~ creating a condition filter in a standard query language statement;

means in the condition filter for defining properties to be satisfied by a result of the query, the defining properties means comprising a tree data structure for storing conditions relating to data values, wherein the data values in the condition filter are replaced with a question mark character and a corresponding data value list is created allowing retrieval of conditions from the tree data structure; and

means in the condition filter for using an object to execute a precompiled standard query language (SQL) statement, wherein the precompiled SQL statement is executed multiple times without being recompiled.

23. (canceled)

24. (canceled)